Designing for balance in a role-playing game

COMP 4560

U6343103
Zhijing Ke
Bachelor

10/6/2020
Abstract

Reaching balance for various gameplay styles inside each particular game is advantageous due to the effective utilizing of software content and the extending of game playability. This project is focusing on a practice of reaching balance between two different gameplay style in a self-developed linear process role-playing game. By using techniques of software engineering, design of experiment and mathematical statistics, the balance between 2 different gameplay styles is well controlled inside both software development stage and balance improvement stage. In the balance evaluation and validation part, those gameplays reached an acceptable equilibrium according to volunteers’ test data, time cost on thinking, and feedback, while unique gameplay features were remained.

Motivation and goals

Inside games which have choices to let players have different gameplay styles, experience and feedback, balance will be significant for the selections provided to game players among those choices. Disadvantaged selections in PVP games will cause unsatisfactory play feedback due to the inferior status caused by those selections in the competition. Selections with good balance design and consideration inside stand-alone game will extend the playability and game software live without additional software content. Reaching good balance between the game choice inside the game is important for both game designers, game developers and even game players.

This project is aiming to reach an equilibrium in a pivotal game process divergence for a self-developed linear process game software, and then evaluate and validate the balance for this equilibrium of the game software.

The process could be roughly divided into 3 sub-goals: software development, balance design and improvement, and final evaluation and validation. In the actual practice the software development and balance improvement are concurrent before doing the final evaluation part.

Model and technique

1) Model outline

This research is based on a simplified strategy game model. The core gameplay is that a series of consequent static play maps will be given to game player. Inside those maps, player can choose their unique route forward to advance the game progress. During choosing and deciding their route forward, player needs to consider the cost they will pay and the benefit they will gain before taking actions, character’s status will also be influenced by those actions.

Cost and payoff are two important concepts for model design, technique practice and balance improving in this research. Inside the model design, cost is the damage of fighting enemies, consuming items. Payoff, or benefit, is the promotion of character’s status and acquisition of expendable items.

The major route divergence, which is also the point need to reach equilibrium, was placed in the middle of the game progress. The model provided 2 different routes:
Warrior and Magician, to game player. In the following of the report those two routes will be called route W and route M. The core design idea for route W is to let player have lower cost when taking actions and doing route selection, while for route M is to let player have higher payoff and better timeliness of the payoff. Both routes are using same play map and have same game process.

Enemy and player will also have special skills to influence cost and mechanism. Some of those skills was created for gameplay while other skills might be created to balance dominant route and dominated route.

2) Mechanism

There are 3 common mechanisms throughout the game: Fighting against enemies, consuming items, and getting benefit. The first two mechanisms decide how much the cost is, and the getting benefit decide the payoff for player’s action. The cost of fighting against enemies is dynamic due to possible change of character’s status and use of special skills, while consuming items could be considered as static cost.

During the fighting, character and enemy have their own attack value (atk), defense value (def) and health point (hp). In each round, they will cause a damage to each other by their attack value minus the other’s defense value. If one’s attack value is not greater than the other’s defense value, no damage will be settled in that round. The fighting will be lasting to a round till one’s damage taken is greater than his own health point. If the player’s health point reduced to 0, play will lose the fight and game will end. Otherwise player will take the damage and continue the game, while enemy will disappear from the play map permanently. Overall, when player win the fight the damage is calculated as:

\[
\text{Damage Per Round (DPR)} = \text{Enemy'}\text{atk} - \text{Player'}\text{def}
\]
\[
\text{Rounds} = \left\lfloor \frac{\text{Enemy}'}{\text{hp}} / (\text{Player}'}\text{at} - \text{Enemy'}\text{def}) \right\rfloor
\]
\[
\text{Damage} = \text{DPR} \times \text{Rounds}
\]

Formula 1 Damage calculation

The other component of the cost is consuming items and resources. The model used “key” and “door” on the map to represent getting and using recourse. To open a closed-door player needs to use a key with corresponding color. Keys in different color have different value.

Getting payoff is that player can collect items and resources from the play map. Items can recover or promote characters’ status. Promoted status of the character will have different cost inside each action selection, which will also influence the further route forward selection.

3) Technique introduction

Inside the software developing, the research used incremental model to transfer the difficult global balance planning to balancing control for single small units. Balance evaluation and validation will be applied to each small sections. By studying how player might take actions during the route selection, player’s tendency can be referred as strategy selection of imperfect information game in the game theory. Some design of experiment and mathematical statistics concepts were used in the design section and evaluation section.
Software introduction

The software is developed on RPG maker engine, with icons and music from free online material library. The product is a linear process RPG game with appropriate balance for both routes of the game. The size of valid play map is 30 and route divergence locates in the 15th map. The total play time is around 3 hours to complete the game with a single route. Tutorial, description, and statement were attached inside the game directory.

Designing for balance

For large-scale online game, balance adjustment on version update is usually based on player’s feedback, big data, group work of data miner and data analyst. While individual game developer or group is not very possible to work with professional data miner and data analyst in the software development and testing stage. One of the mainstream solutions for balance issue is that using internal test with volunteers and make open beta version of the software for public test, then adjust and remake the balance with negative feedback according to test data.

In this research, volunteers’ feedback and save file data is a major basis to decide where need to be improved to reach better balance. By comparing the resources change,
time spend on thinking, and volunteers’ opinion toward the difficulty, the area or range need to apply balance strategy could be determined. According to the result, one of the routes of route M and route W will be defined as dominant route and the other will be defined as dominated route. This concept is like strategy provided to player inside game theory: overall dominant route will always get higher payoff with lower cost almost everywhere while dominated route might get relatively lower payoff with higher cost.

This research designed and tried 2 major method to control the balance in the operation, which are from software engineering level and from strategy and practice level.

1) **Software engineering level**

The core idea for software engineering level strategy is divide and conquer. Before new maps, new skills or items are deployed, volunteer group will test those new objects as several single units. Considering too much content in one test will increase the difficulty of balance test and control, and too little content might cause split of the gameplay and homogenization between play maps, the size of each incremental element was set as 5 new play maps.

2) **Strategy and practice level**

The strategy and practice are for balance adjustment and will be applied to the unbalanced part discovered by balance test. In this research, the 3 most common strategies and practices for balance adjustment are: remake of the play map, adjustment of numerical value, remake of design and mechanisms. These 3 strategies have a priority from high to low in the actual practice. The main idea for all strategy and practice is to support the dominated route and weaken the advantages of dominating route.

The first method is the remake of play map. By raising the cost or reducing the payoff, the expected benefit and difficulty will be controlled. This operation usually locates in a single play map or even in one area, which is safe for the global balance situation.

The second method is the adjustment for numerical value inside the game. It is controllable if the changed numerical is not a global value. These adjustments usually locate in value for single items and or a kind of enemies in one incremental element, which is also considered controllable for the global balance.

The third method is the change of mechanisms and design. This is a challenging work in most of the time. The remake of mechanisms is very possible to break the balance status reached in former stages. While a safer way to complete the mechanisms’ change is to create an independent item, skill or enemy and just apply the enemy inside where the game balance is insufficient. Reuse those additional content if it is necessary in the following developing.

In the actual practice, the first method had the highest frequency in the actual balance adjustment, while the other 2 methods are not so frequent.

**Result and Outcome**

The final balance evaluation and validation used blind test with volunteers not in previous test group. The balance of route M and route W will also be evaluated by resources remaining when complete the game, time spending during playing and their
opinion. Each route will have 5 volunteers to play the route undetermined.

The result statistics shows that volunteers playing route W had higher satisfaction, lower average game completing time and higher completing rate.

Both routes have a pleasant complete rate. The route M have higher variance for complete status and time cost.

By comparing the actual route and action selection, route W tended to converge in some pivotal choices. In contrast, the route M’s selection and usage of item was more differentiated.

<table>
<thead>
<tr>
<th>Route</th>
<th>Complete rate</th>
<th>Average time cost</th>
<th>Difficulty feedback</th>
<th>Recourse consuming</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>100%</td>
<td>42.0 min</td>
<td>40%</td>
<td>80%</td>
</tr>
<tr>
<td>W</td>
<td>60%</td>
<td>63.3min</td>
<td>60%</td>
<td>40%</td>
</tr>
</tbody>
</table>

Table 1 Statistics of average complete status.

![Blind test group statistics](image)

Figure 3 Histogram for route M and W’s statistics

**Conclusion**

Considering the statistics of the final balance evaluation and validation, the software reached a balance of those two routes to some extend: Two routes have different difficulty while the difficulty difference had not influenced the balance situation significantly. Skillful route had higher upper bound and onerous lower bound, the play feedback and complete situation of this route was more differentiated. At the same time the smooth route had similar route selection, time cost and play feedback. Besides, the capacity of the volunteer group is not very sufficient, which might affect the correctness.
of the conclusion about balance status. In further research, balance inside multiple gameplay routes could be designed and studied, and volunteer data can be more sufficient for a more precise conclusion.

References

Appendix
Appendix 1 Project Description

In this project, student will develop a role-playing game and implement a model to ensure game balance according to game design theory and volunteers’ feedback. Development will be based on a simple role-playing game framework, which was implemented on RPG maker engine.

In the project, student will focus on:
1. The balance of the game based on game design theory. Ensuring balance on significant choices inside the game.
2. Design implementation: Designing and implementing skills, maps and items of the game.

Expected outcome:
A well programmed game with reasonable balance between target game choices. Evaluation and study about how and why the researched reached the balance.

Appendix 2 Study Contract
INDEPENDENT STUDY CONTRACT
PROJECTS

Note: Enrolment is subject to approval by the course convenor

SECTION A (Students and Supervisors)

UnitID:       _u6343103___
SURNAME:      Ke__________________ FIRST NAMES:     Zhijing__________________
PROJECT SUPERVISOR (may be external):       Penny Kyburz__________________
FORMAL SUPERVISOR (if different, must be an RSICS academic):       Penny Kyburz__________________
COURSE CODE, TITLE AND UNITS:    COMP4560_____12units__________________

COMMENCING SEMESTER:  ☑ S1 ☐ S2 YEAR:  2020. Two-semester project (12u courses only):  ☐

PROJECT TITLE:
Designing for balance in a role-playing game

LEARNING OBJECTIVES:
1) Developing skills in game design, process, and development
2) Applying and improving coding skills for game development
3) Applying game design theory to developing a game

PROJECT DESCRIPTION:
In this project, I will develop a role-playing game and implement a model to ensure game balance according to game design theory and volunteers' feedback. Development will be based on a simple role-playing game framework, which was implemented on RPG maker engine.

In the project, I will focus on:
1. The balance of the game based on game design theory.
   Ensuring balance on significant choices inside the game.
2. Design implementation.
   Designing and implementing skills, maps and items of the game.

The following is the process of the project:
1. Game development (24 Feb to 17 Apr)
   Building the game maps, skills, items and enemies.
2. Game improvement. (20 Apr to 1 May)
   Debugging, balancing skills and items according to feedback.
3. Game validation. (4 May – 22 May)
   Test of the game balance by volunteers.
4. Presentation. (25 May to 29 May)
   Submitting of the game software, report writing and presentation preparing.

Research School of Computer Science

Form updated Jun 2018
Appendix 3 & 4 Readme File

This software is developed with RPG maker engine, with running environment as windows 10, i7 6930. Please Click and run the "game.exe" in the root directory with orange icon to start.

The code and script implemented by the student was placed and introduced in the file "Implementation.doxo", which include the map design, part of the skills' and items' design and implementation. Mechanism design and implementation.
Basic operation and software introduction were placed in the file "Tutorial.txt".

Statement and acknowledgements:

All the video, music, graphics material, inbuild script were from free online resource library like RPGmaker forum and 66rpg.com.
The menu interface, game display, map editor, conversation recorder, animation scripter are original files from RPGmaker forum.
Part of Battle mechanism, damage calculation, skill mechanism and usage of item are the modification based on existing scripter.

The software was tested can could be run in Windows 7, windows 10. Some bugs like "cannot find file xxx" might be solved by add the font inside root directory with Chinese font required. If it still cannot work, please contact with u6343103@anu.edu.au
The test result for final balance evaluation is recorded in the file "Record.txt"

Thanks for the reading!